

VIR.US.EXE

project by carlos katastrosky
text by Luís Silva

- + Computer viruses have long been considered our machines' most fearsome foes. Able to replicate themselves and spread uncontrollably through communication networks, they constitute a direct threat to all the information we keep in our disks and hold so dear. But computer viruses, like their organic counterparts, function, even if we tend to perceive it as one single effect, on two different levels. First of all, they *infect*, their rogue activity has consequences, damaging the hosting systems and spreading their reach; and secondly, they also *threat*, causing a generalised feeling of insecurity and fear, originated by the perception of the eventual infection's outcomes.

carlos katastrosky's project *vir.us.exe* (2009), co-commissioned by KURATOR and LX 2.0 for the *Anti-Bodies* programme, departs from and investigates this dual nature that defines viral activity. *vir.us.exe* is a windows program that once downloaded by the user and executed in her machine, will simply delete itself. Despite such a harmless and even self-destructive behaviour (on the antipodes of a common viral infection), the program is defined and promoted as an actual virus, that one, if brave or careless enough, can install, risking her computer and in the process compromising her own security.

Reminiscent of *Russian Roulette* (2006), a p2p-related piece in which a call was launched inviting users to upload files of their choice that could later be randomly downloaded by other users who had no knowledge whatsoever of the content of

those files, possibly threatening the integrity of their machines, *vir.us.exe* is, like many other katastrosfsky projects, a silent project. One of two things can happen: either the user perceives *vir.us.exe* as an actual threat and doesn't engage with the application, maintaining her system safe; or she will download and run the application, leading to its self deletion, and as in the previous situation nothing much will happen. Or will it? The core of the project doesn't lie in running or not running a piece of software in order to obtain a certain outcome. The application is simply an excuse, a set-up that is carefully created by the artist to trigger a response in the user, to confront her and investigate how the psychology of fear works (similar to the strategies and methodologies of early social psychology experiments) and causes one to react. *vir.us.exe* isn't a virus because it infects a computer, it is a virus because it triggers the exact same responses every virus triggers, regardless of causing an actual infection. In that sense we can say it becomes a meta-virus, not threatening in itself, but being perceived as a threat.

↻

KURATOR <<http://www.kurator.org/>> is collaborating with LX 2.0 <<http://www.lisboa20.pt/lx20/>> to develop a programme of new work to infect the 2012 Olympics. The overall *Anti-Bodies* programme is curated and co-ordinated by Relational <<http://relational.org.uk/>>, with support from Arts Council England.

```
#include <windows.h>
#include <shellapi.h>
#include <iostream>

void Selfdestruct();
static const char tempbatname[] = "_uninsep.bat" ;

void main(void)
{
    Selfdestruct();
}

void Selfdestruct()
{
    // temporary .bat file

    static char templ[] =
        ":Repeat\r\n"
        "del \"%s\" \r\n"
        "if exist \"%s\" goto Repeat\r\n"
        "del \"%s\" " ;

    char modulename[_MAX_PATH] ;    // absolute path of calling .exe file
    char temppath[_MAX_PATH] ;    // absolute path of temporary .bat file
    char folder[_MAX_PATH] ;

    GetTempPath(_MAX_PATH, temppath) ;
    strcat(temppath, tempbatname) ;

    GetModuleFileName(NULL, modulename, _MAX_PATH) ;
    strcpy(folder, modulename) ;
    char *pb = strrchr(folder, '\\');
    if (pb != NULL)
        *pb = 0 ;

    HANDLE hf ;

    hf = CreateFile(temppath, GENERIC_WRITE, 0, NULL,
        CREATE_ALWAYS, FILE_ATTRIBUTE_NORMAL, NULL) ;

    if (hf != INVALID_HANDLE_VALUE)
    {
        DWORD len ;
        char *bat ;

        bat = (char*)malloc(strlen(templ) +
            strlen(modulename) * 2 + strlen(temppath) + strlen(folder)
            + 20) ;

        wsprintf(bat, templ, modulename, modulename, folder, temppath) ;

        WriteFile(hf, bat, strlen(bat), &len, NULL) ;
        CloseHandle(hf) ;

        ShellExecute(NULL, "open", temppath, NULL, NULL, SW_HIDE);
    }
}
```